上級編

# 第 34 講 コインの枚数

## 例題

銀行の ATM である金額を引き出すとき，紙幣・硬貨の枚数の合計が最も少なくなるように紙幣・硬貨がそれぞれ何枚必要かを表示するプログラムを作成する。

例えば，12,346 円を引き出す場合は以下のように表示される。

実行結果

```
10000 円 : 1 枚
5000 円 : 0 枚
1000 円 : 2 枚
500 円 : 0 枚
100 円 : 3 枚
50 円 : 0 枚
10 円 : 4 枚
5 円 : 1 枚
1 円 : 1 枚
```

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
Okane = [10000, 5000, 1000, 500, 100, 50, 10, 5, 1]
Maisuu = [0, 0, 0, 0, 0, 0, 0, 0, 0] # 引き出すお金の枚数を保持（Okane 配列の添字と対応）
okane_count = 0 # お金の枚数を一時的に保持

hikidashi = int(input('引き出し金額を入力してください：'))

for i in range(len(Okane)):
  okane_count = [          ]
  Maisuu[i] = okane_count
  hikidashi = hikidashi % Okane[i]

# 表示処理
for i in range(len(Okane)):
  print(Okane[i],'円：',Maisuu[i],'枚')
```

① hikidashi * Okane[i]

② hikidashi % Okane[i]

③ hikidashi ^ Okane[i]

④ hikidashi // Okane[i]

# 上級編 第 34 講 確認問題

以下の条件で予算を入力すると，いくらのお菓子を何個買えるか表示するプログラムを作成する。

- 350 円，170 円，90 円，55 円，30 円，20 円 の 6 種類のお菓子が売っている商店で買い物をする。
- できるだけ高いお菓子をたくさん買い，予算をできるだけ使い切る買い方をする。
- 配列 Okashi には，お菓子の値段が降順に格納されている。

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
Okashi = [350, 170, 90, 55, 30, 20] # 売っているお菓子の金額 ( 値段の降順 )
Kosuu = [0, 0, 0, 0, 0, 0]

kingaku = int(input(' 予算を入力してください：'))

for i in range(len(Okashi)):
  Kosuu[i] = kingaku // Okashi[i]
  kingaku = □

for i in range(len(Okashi)):
  print(Okashi[i],' 円のお菓子：',Kosuu[i],' 個 ')
```

① `Kosuu[i] * kingaku`

② `kingaku * Okashi[i]`

③ `Okashi[i] % kingaku`

④ `kingaku % Okashi[i]`

# 上級編　例題　解答

## 第34講　コインの枚数

例題　解答：④

## 第35講　駐車料金の判定

例題　解答：④

## 第36講　割引のある料金

例題　解答：④

## 第37講　テストの集計

例題　解答：①

## 第38講　じゃんけん

例題　解答：①

## 第39講　サイコロの出目

例題　解答：③

## 第40講　中央値の判定

例題　解答：①

## 第41講　素数判定

例題　解答：③

## 第42講　素因数分解

例題　解答：②

## 第43講　倍数の判定・FizzBuzz

例題　解答：④

## 第44講　バブルソート・選択ソート

例題　解答：③

## 第45講　グラフ理論

例題　解答：②

## 第46講　待ち行列

例題　解答：②

## 第47講　パリティチェック

例題　解答：①

## 第48講　すごろくゲーム

例題　解答：④

# 上級編　第 34 講　確認問題　解答

正解：④

解説：

7 行目の Kosuu[i] = kingaku // Okashi[i] では，配列 Okashi の i 番目のお菓子の購入個数を配列 Kosuu[i] に代入している。そのため，続く 8 行目では，配列 Okashi の i 番目のお菓子を購入したあとの残金を，変数 kingaku に代入すればよいと考えられる。残金は，現在の金額から i 番目のお菓子の金額を購入した余りの金額であるため，空欄には，kingaku % Okashi[i] が当てはまる。よって，正解は④ kingaku % Okashi[i] である

```
01 Okashi = [350, 170, 90, 55, 30, 20] # 売っているお菓子の金額（値段の降順）
02 Kosuu = [0, 0, 0, 0, 0, 0]
03
04 kingaku = int(input('予算を入力してください：'))
05
06 for i in range(len(Okashi)):
07   Kosuu[i] =  kingaku // Okashi[i]
08   kingaku = kingaku % Okashi[i]
09
10 for i in range(len(Okashi)):
11   print(Okashi[i],'円のお菓子：',Kosuu[i],'個')
```

実行結果

```
予算を入力してください：3334
350 円のお菓子： 9 個
170 円のお菓子： 1 個
90 円のお菓子： 0 個
55 円のお菓子： 0 個
30 円のお菓子： 0 個
20 円のお菓子： 0 個
```